Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Secure semantic search using deep learning in a blockchain-assisted multi-user setting

Shahzad Khan[1,2*], Haider Abbas[1] and Muhammad Binsawad[3]

Abstract

Deep learning-based semantic search (DLSS) aims to bridge the gap between experts and non-experts in search. Experts can create precise queries due to their prior knowledge, while non-experts struggle with specific terms and concepts, making their queries less precise. Cloud infrastructure offers a practical and scalable platform for data owners to upload their data, making it accessible to intended data users. However, the contemporary single-owner/single-user (S/S) approach to DLSS schemes falls short of effectively leveraging the inherent multi-user capabilities of cloud infrastructure. Furthermore, most of these schemes delegate the dissemination of secret keys to a single trust point within the mutual distrust scenario in cloud infrastructure. This paper proposes a Secure Semantic Search using Deep Learning in a Blockchain-Assisted Multi-User Setting (S^3DBMS). Specifically, the seamless integration of attribute-based encryption with transfer learning allows the construction of DLSS in multi-owner/multi-user (M/M) settings. Further, blockchain's smart contract mechanism allows a multi-attribute authority consensus-based generation of user private keys and system-wide global parameters in a mutual distrust M/M scenario. Finally, our scheme achieves privacy requirements and offers improved security and accuracy.

Keywords Blockchain-based verification, Semantic search, Multi-authority attribute-based encryption, Secure transfer learning, Attribute based access control

Introduction

To reduce costs linked with managing large amounts of data, more individuals and organizations are choosing to entrust the management of this data to the public cloud. To maintain the privacy and security of the data being entrusted, it is a usual practice to encrypt the data before sending it to the cloud service. When dealing with this outsourced encrypted data in a cloud infrastructure, the primary and essential action involves initiating a search process. This search serves as the principal entry point for reaching and interacting with the encrypted data in the cloud environment before any subsequent tasks or analyses can be conducted.

Deep learning natural language processing (NLP) models have revolutionized information retrieval, making semantic-aware searches accessible even to those without specialized knowledge in a particular field. These models operate through complex neural networks, which enable them to understand the nuances of human language beyond the confines of traditional keyword-based approaches. Unlike conventional search algorithms that rely solely on exact keywords, deep learning NLP models go a step further by considering the context and intent behind user queries. This nuanced approach allows these models to provide more sophisticated and accurate search results, enhancing the overall user experience. For individuals lacking expertise in a specific domain, the advent of deep learning NLP models is particularly beneficial. Users can now formulate queries in a natural and conversational manner, eliminating the need

*Correspondence:
Shahzad Khan
skhan.phdismcs@student.nust.edu.pk; shehzad@sbbu.edu.pk
[1] University of Science and Technology, NUST, 44000 Islamabad, Pakistan
[2] Department of Computer Science, Shaheed Benazir Bhutto University Sheringal, Dir (Upper) 18000, Pakistan
[3] King Abdulaziz University, 21589 Jeddah, Saudi Arabia

for meticulously crafted keyword-heavy requests. The models, in response, excel at understanding user intent, leading to more precise and relevant search outcomes. However, contemporary schemes integrating deep learning-based models into searchable encryption(SE) predominantly operate within single-owner/single-user settings. This prevalent approach significantly restricts the adaptability and versatility of cloud storage solutions.

Therefore, in this paper, our novel objective is to introduce deep learning-driven semantic-aware searchable encryption (SE) in a multiple-owner/multiple-user (M/M) setting. This is realized through the integration of attribute-based encryption (ABE) with secure transfer learning. However, employing this novel insight in the M/M setting is a non-trivial task. Particularly, we need to properly handle the following two key challenges. To start with, Attribute-Based Encryption (ABE) is a cryptographic primitive that provides a fine-grained access control mechanism for encrypted data. It enables data owners to encrypt their data in a way that only users possessing specific attributes can decrypt and access it. This is particularly valuable in scenarios where access control requirements are intricate and dynamic. However, the traditional ABE scheme inherently operates on a trusted attribute authority (TAA), as shown in Fig. 1, which contradicts the basic premise of a multi-user setting in the mutually distrustful scenario of cloud infrastructure. This situation leads to scalability, availability, privacy, and trust issues. Therefore, a secure distributed mechanism is desired to be placed on top of the traditional ABE primitives while still maintaining access at a fine granularity. Second, in a scenario where there are multiple data owners and multiple data users, all operating in an environment of mutual distrust (such as cloud infrastructure), a revocation mechanism becomes crucial for maintaining the security and integrity of the system. A revocation mechanism is a way to handle situations where access to certain data must be revoked or invalidated for specific users or groups. A revocation mechanism becomes even more important in the context of Attribute-Based Encryption (ABE), which provides fine-grained access control based on attributes. In a scenario where a single point of trust is absent, direct revocation remains the primary option. Here, the responsibility for specifying the revocation list during the encryption process falls directly upon the data user.

In the M/M setting, an extra demand arises: the data owners (DOs) need to be aware of who accessed their outsourced data and when, while the data users (DUs) must have confidence that the accessed data remains unaltered. We construct a blockchain-assisted multi-attribute authority scheme to ensure the privacy of deep-learning-based semantic-aware searches and address the demands of a multi-data owner and multi-data user (M/M) setting within a scenario of mutual distrust.

This study's main contributions can be summarized as follows:

1. We leverage the capabilities of blockchain's smart contract mechanism to establish a multi-attribute

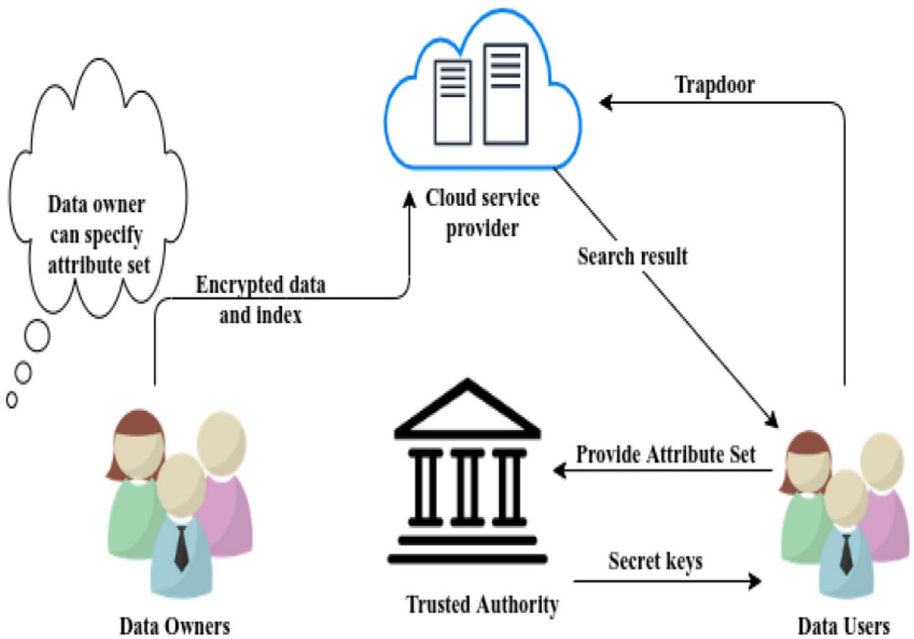

Fig. 1 Attribute-based searchable encryption

authority SE scheme. This integration of smart contracts avoids the dependence on a singular trusted entity within an ABE infrastructure. Instead, it facilitates the consensus-based generation of user private keys and system-wide global parameters in a mutual distrust scenario M/M setting.
2. We seamlessly integrated user revocation into the ciphertext managed by data owners. This integration naturally accommodates the absence of a singular trust point within the ABE mechanism while preserving the unaffected status of non-revoked users.
3. We combine deep learning-based transfer learning with Semantic Term Matching Constraints (STMC) to achieve search results with high accuracy and ranking. This ensures that the index and query feature vectors have an identical feature space and underlying distribution.
4. Our approach employs a smart contract to record public system parameters and user actions, covering data upload, search, and download processes. Blockchain's inherent features ensure that operation records are tamper-resistant, allowing data owners to monitor data access following the upload process easily.

Related work

Secure semantic search schemes

Semantic search aims to bridge the gap between experts and non-experts in search. Experts can create precise queries due to their deep knowledge, while non-experts struggle with specific terms and concepts, making their queries less precise. Various strategies have been developed to achieve secure semantic searching, including methods like query expansion and word embeddings. Expansion-based approaches involve extending query terms to include more words that are semantically relevant. This increases the likelihood of matching predefined keywords effectively. The techniques in this category are broadly categorized into three groups: mutual information, synonyms, and concept hierarchies. As demonstrated by [1, 2], mutual information-driven techniques utilize the probabilities of hashed keyword co-occurrences. They create a mutual information model to expand hashed query terms within the encrypted data. The majority of synonym-oriented methods [3, 4] and strategies based on concept hierarchies [5, 6] revolve around extending query terms in the plaintext. These techniques subsequently construct secure indexes utilizing the SecKnn algorithm [7]. Word embedding is a methodology for representing words in a vector format to retain their semantic context. Yang and Zhu [8] utilize word embeddings to introduce a secure semantic search approach employing linear optimal matching, leading to accurate search outcomes. Additionally, [9] presents an embedding-centered scheme wherein documents and queries are transformed into condensed vectors. This scheme then employs the SecKnn algorithm to encrypt these compact vectors. These strategies tackle the issue of semantic-aware absence, yet they remain confined to single-data owner (DO) or single-data user (DU) setups. This limitation significantly curtails their applicability across a wider spectrum of real-world scenarios.

Verifiable blockchain-based schemes

Ensuring verifiably secure searching entails the integration of verification mechanisms within schemes. These mechanisms serve to validate the accuracy of search outcomes without compromising the confidentiality of sensitive information. The majority of verifiable searching techniques [10–12] utilize methodologies such as the Merkle Hash Tree or other variants to establish an anticipated checklist. This checklist serves the purpose of cross-validating the search results for accuracy without compromising sensitive information. Yang and Zhu [8] research presents a novel verification approach that uses intermediate search data to validate the accuracy of search results. Nevertheless, these strategies presuppose the involvement of a trusted entity to oversee the verification process. For instance, certain approaches rely on methods centered around a trusted third party. However, this approach raises noteworthy concerns and challenges pertaining to both security and operational efficiency [13]. In the context of data management, certain schemes [13–15] adopt blockchain-based public audits. These audits serve to mitigate the reliance on single points of trust. Furthermore, certain research studies employ blockchain's unchangeable and consensus nature to create reliable methods for detecting accurate information and establishing resilient information tracking systems [16, 17]. The multiparty transaction scheme introduced in [18] is built upon blockchain technology. In this approach, all pertinent details concerning a transaction are consolidated within a single block. This design enhances the efficiency of ledger traceability and audit processes. The approach in [19] integrates the advantages of attribute-based cryptography and the chameleon hash function, successfully attaining key security features while maintaining a high level of efficiency. Alternative blockchain-driven approaches [20–23] create smart contracts for their individual retrieval processes on the blockchain. For instance, [20] suggests an encrypted index-based, privacy-focused decentralized storage system where blockchain nodes perform on-chain retrieval to agree on search outcomes. Additionally, [24] introduces a two-layer verification mechanism.

Users conduct the initial verification using a checklist to ensure accurate search results, followed by blockchain nodes performing a second verification by re-executing on-chain retrieval to establish consensus on the outcomes. The mentioned studies share two main drawbacks. Firstly, they are limited to a single-secret key (S/S) setup, allowing only the key holder to create search queries, restricting the search scope. Secondly, none of these approaches incorporate revocation in the dynamic and scalable cloud infrastructure architecture.

Notations and background knowledge

Notations

- $\mathbb{G}_1, \mathbb{G}_T$ — Bilinear groups of order p.
- $\mathcal{K}_{local}, \mathcal{DK}$ — Data user's local key and delegated key respectively.
- $\mathbb{Z}_p$, $\mathbb{Z}_p^*$ — Finite fields, whose integer elements are $\{0, 1, \ldots, p-1\}$ and $\mathbb{Z}_p \setminus \{0\}$ respectively.
- U — Universal set of attributes.
- γ — DO chosen set of attributes.
- S — DU set of attributes.
- D — The plaintext set of n documents, namely $D = \{D_1, D_2, \ldots, D_n\}$.
- $\tilde{D}$ — The encrypted set of n documents, denoted by $\tilde{D} = \{\tilde{D_1}, \tilde{D_2}, \ldots, \tilde{D_n}\}$.
- D_v — The index document vector generated from D by the Doc2Vec model, denoted by $D_v = \{D_{v_1}, D_{v_2}, \ldots, D_{v_n}\}$.
- I — The encrypted index vector based on D_v.
- m — Number of features in the Doc2Vec generated vector.
- Q — The search query keywords set, which is denoted as $Q = \{q_1, q_2, \ldots, q_n\}$.
- Q_v — The query feature vector generated from search query Q by Doc2Vec model.
- $\tilde{Q_v}$ — The trapdoor vector based on query feature vector.
- M_w — Extracted Doc2Vec hidden layer weights matrix.
- $\tilde{M}_w$ — Secure weights matrix based on M_w.
- M_1, M_2 — Invertable matrices for inner product operation.
- $\tilde{M}_1, \tilde{M}_2$ — Secure invertable matrices based on M_1 and M_2 respectively.
- $RoTR(x)$ — Circular right shift notation of an argument x.
- $LoTR(x)$ — Circular left shift notation of an argument x.
- A_c — Access control.
- V — Configuration vector.

Preliminaries

In this section, an overview of cryptographic primitives and feature-extraction techniques is presented.

Bilinear pairing

Let $\mathbb{G}$, $\mathbb{G}_T$ be two cyclic groups of prime order p, g be a generator of $\mathbb{G}$, and $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ be the bilinear map which has several properties: (1) $\forall u, v \in \mathbb{G}, a, b \in \mathbb{Z}_p^*, e\left(u^a, v^b\right) = e(u, v)^{ab}$, (2) $e(g, g) \neq 1$, (3) e can be efficiently computed.

Non-monotonic access structure

We recall the definition of a non-monotonic access structure proposed by Ostrovsky et al. [25]. For a set of parties P, a collection of monotonic access structures A has the following properties: either the name is normal (x), or it is primed (x') and if $x \in P$ then $x' \in P$ and vice versa. We write $\breve{x}$ to denote a parity in P that may be primed or unprimed. Conceptually, prime attributes represent the negation of unprimed attributes. For an access structure, $\mathbb{A} \in A$ over a set of parties P, the corresponding non-monotonic access structure is $NM(\mathbb{A})$ over a set of parties $\tilde{P}$, where $\tilde{P}$ is the set of all primed parties in P. Then, we can define the following family $\tilde{\mathbb{A}}$: For every set $\tilde{S} \subset \tilde{P}$ we define $N(\tilde{S})$ as $N(\tilde{S}) = \tilde{S} \cup \{x' \mid x \in \tilde{P} \setminus \tilde{S}\}$. Then, we define $NM(\mathbb{A})$ by saying that $\tilde{S}$ is authorized in $NM(\mathbb{A})$ if and only if $N(\tilde{S})$ is authorized in $\mathbb{A}$.

Semantic term matching constraints

The work of [26–28] has formulated the retrieval constraints on retrieval framework for enhanced retrieval performance. We define D, D_1, and D_2 to represent the documents and Q, Q_1, and Q_2 to represent the queries. Let M and N be parts of the keywords and words in the document and query, respectively. Let $s(k, g)$ be any given semantic similarity between document keyword g and query word k. We assume that the word k is semantically more similar to word g than to word h if and only if $s(k, g) > s(k, h)$. If we let the weight of keyword g in document D as $\tau(g, D)$, then we can define $\mathcal{H}(Q, D)$ the relevance score between query Q and document D. To simplify the explanation, we use $\chi(N, g)$ to represent the semantic similarity between words N and the keyword g and $\mathcal{P}(Q, M)$ to represent the semantic similarity between query Q and keywords M.

STMC-1: For all Q, D_1, and D_2, if Q is composed of k and N, D_1 is composed of g and M, D_2 is composed of h and M, $\mathcal{P}(Q, M) = 0, \chi(N, g) = \chi(N, h) = 0, \tau(g, D_1) = \tau(h, D_2)$, and $S(k, g) > S(k, h)$, then $\mathcal{H}(Q, D_1) > \mathcal{H}(Q, D_2)$.

Within the context of *STMC*-1, the retrieval function necessitates the allocation of higher score to a

document exhibiting a keyword that possesses greater semantic relatedness to the query word.

STMC-2: For all Q, D_1, and D_2, if Q is composed of k and N, D_1 is composed of g and M, D_2 is composed of h and M, $\mathcal{P}(Q,M) = 0$, $\chi(N,g) = \chi(N,h) = 0$, and k equals g, then $\mathcal{H}(Q,D_1) > \mathcal{H}(Q,D_2)$ even $\tau(h,D_1)$ is much smaller than $\tau(h,D_2)$.

Within the context of *STMC*-2, their retrieval function necessitates that the exact match of an original query term Q should consistently constitute a relevance score that is not inferior to the matching of a semantically related term D, irrespective of the frequency of occurrence of term D within the document.

STMC-3: For all D, Q_1 and Q_2, if Q is composed of $\{k,h\}$ and N, Q_2 is composed of k and D is composed of g and M, $\mathcal{P}(Q_1,M) = \mathcal{P}(Q_1,M) = 0\chi(N,g) = 0$ and $S(h,g) = S(k,g)$, then $\mathcal{H}(Q_1,D) > \mathcal{H}(Q_2,D)$.

Within the context of *STMC*-3, the retrieval function necessitates the involvement of greater variety of query words during the search process.

Prediction based embedding

Word embedding makes it possible to represent a language's text using a vector easily readable by the machine, the basic idea behind Natural Language Processing. To preserve the latent semantic relationship between words, Mikolov et al. [29] proposed a Word2Vec vectorization model. Doc2Vec is an extension of Word2Vec that vectorizes an entire paragraph, an article, or a document instead of individual words. Similar to Word2Vec, it can operate in either of its training methods: Distributed Memory Version of paragraph vector (PV-DM) or Distributed Bag of Words Version of Paragraph vector (PV-DBOW) as depicted in Fig. 2(a) and (b) respectively.

Shamir secret sharing scheme

The Shamir Secret Sharing (SSS) scheme enables secure distributed information sharing. Its core concept involves breaking a secret value s into n shares, with secret reconstruction requiring possessing at least k shares. Specifically, in the process of Shamir Secret Sharing (SSS), where s is assumed to be an element of the integer modulo p, and with k representing the threshold and n participants, the procedure unfolds as follows: Initially, a randomly generated polynomial of degree (k-1) denoted as $f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{t-1}x^{t-1}$ is employed, where a_0 is set to s, and a_1, a_2, through a_{t-1} are chosen at random from Z_p. Following this, the algorithm computes n distinct points on the curve defined by this polynomial, and each participant is provided with one of these points, facilitating secure information sharing. The secret s can be restored through the utilization of Lagrange interpolation, which involves a selection of k out of the n available points in accordance with the equation $s = \sum_{i=1}^{k-1} y_i \prod_{\substack{m=0 \\ m \neq j}}^{k-1} \frac{x_m}{x_m - x_j}$.

System model

Architecture

Our system model architecture is shown in Fig. 3, which consists of five entities. The detailed characteristics and functions of each entity are introduced as follows.

MAA: MAAs (Multiple attribute authorities) are responsible for providing attributes to data users using blockchain. Within MAAs, various attribute authorities contribute to the pool of attributes. This is crucial for ABE schemes, as they're used in distributed access control environments where a single authority might not know all attributes. A distinct authority manages each attribute.

Blockchain: In the situation we are examining, the shared data contains personal information. This is why we're implementing a Consortium Blockchain in our scheme. Considering the underlying scenario, respected and well-established institutions will serve as consensus nodes. Their role is to manage the blockchain and create blocks of information. In parallel, smart contracts facilitate attribute authorities in configuring overall settings and generating secret keys for users.

DO: DO encrypts documents for outsourcing to CS, allowing authorized Data Users (DU) to perform semantic searches. DO creates feature vectors by training a Doc2Vec model on the document set, securing model weights. Encrypted documents and index are outsourced to the CS.

DU: Data Users (DU) interact with the Cloud Storage (CS) to perform searches using encrypted indexes. The DU's identity is verified through attribute-based access control. If the DU meets the access criteria set by the DO, the CS provides encrypted components $\tilde{CT}$. The DU employs the Doc2Vec model, loading it with query keywords from set Q to generate the query feature vector Q_v. The encrypted vector $\tilde{Q}_v$ is then submitted to the CS to retrieve the top-k relevant documents.

CS: It manages storage and search queries for DO and DU, respectively. The Cloud Storage (CS) communicates twice with DU: first to confirm access permissions based on attributes and then to retrieve the top-k relevant encrypted documents from secure indexes in response to search queries.

Workflow of *S*^{3}*DBMS*

The diagram in Fig. 4 illustrates the sequence of steps in our proposed scheme. To begin, the security parameter is inputted to the attribute authorities and the blockchain. They then employ Shamir's secret scheme for N attribute authorities to create the global system parameters. After that, data users enroll themselves

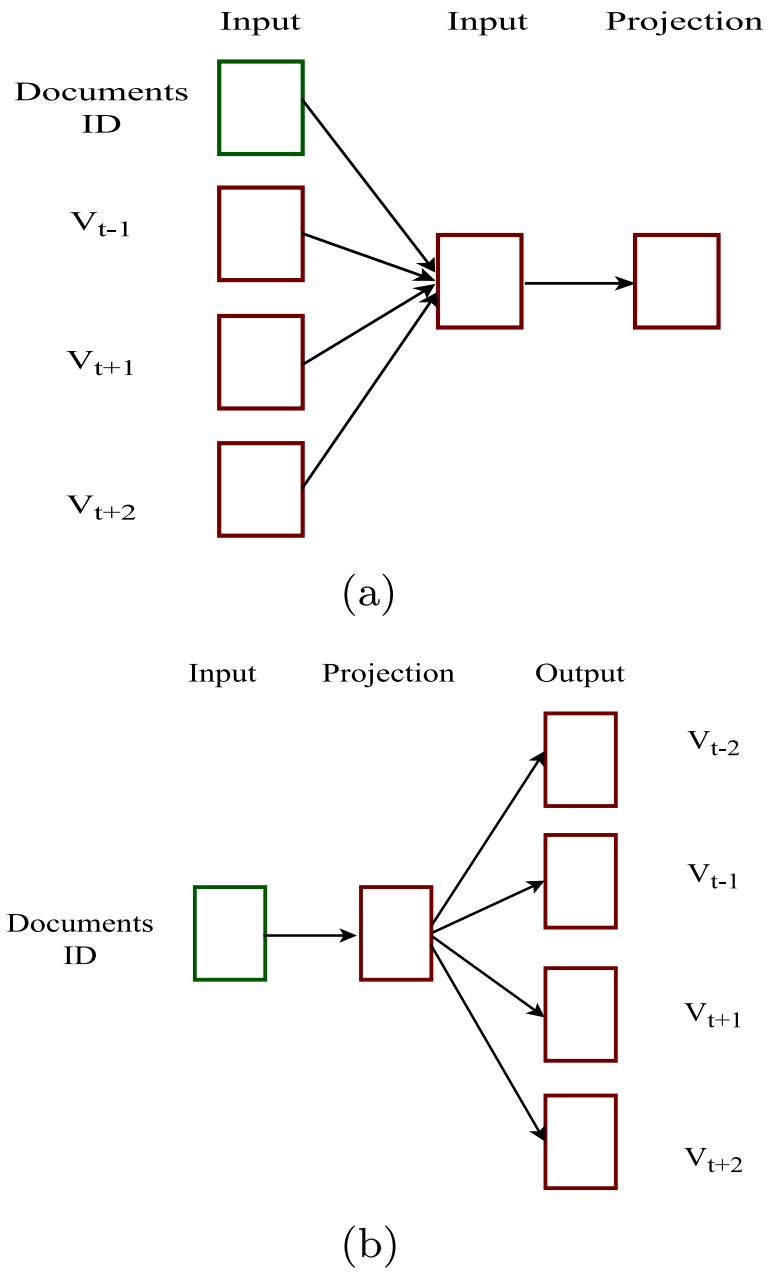

Fig. 2 Doc2Vec framework for (**a**) PV-DM (**b**) PV-DBOW

with the blockchain by utilizing the registration public key RPK_{uid}, which is generated using the system parameters. Next, the multi-attribute authorities utilize the DU set of attributes and system parameters along with their registration key to construct the partial delegated key PDK_{uid}. Subsequently, the blockchain's smart contract calculates the complete secret key SK_{uid} for the DU through Lagrange polynomial interpolation.

Furthermore, DO encrypts their documents using system parameters, creating CT. This encrypted content is stored with a CS. CT includes encrypted documents, secure indices, a revocation list (RL), and access control elements (A_c). The CS maintains the encrypted document and logs the encryption process through a blockchain-based smart contract. Subsequently, the DU submits the delegated key DK_{uid} to the CS. After

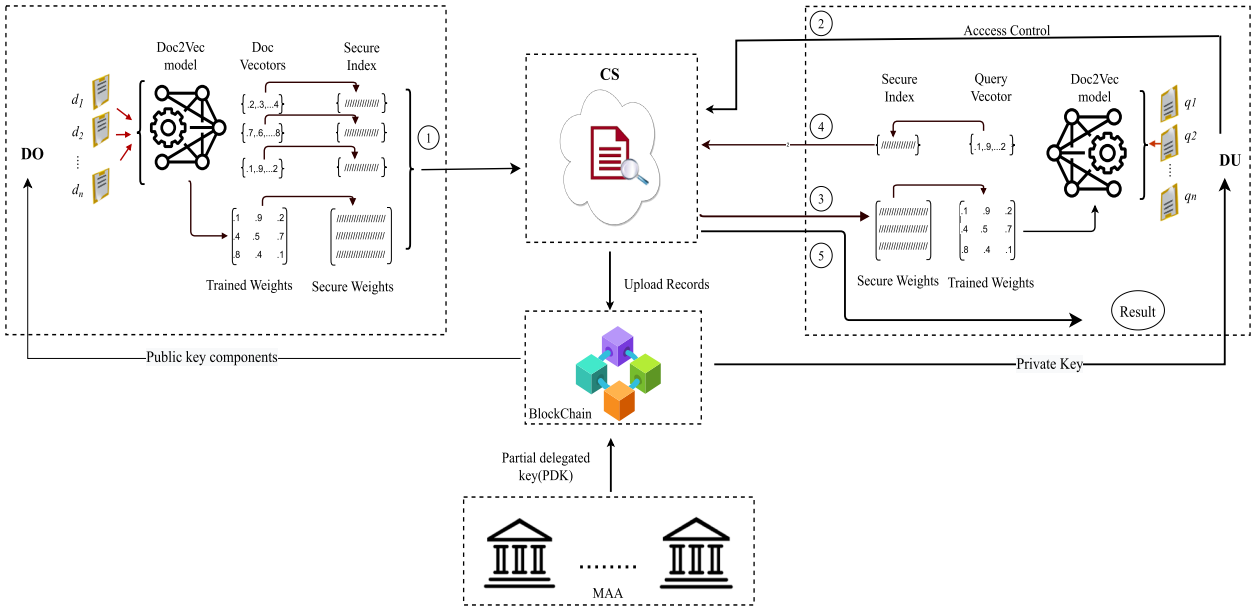

Fig. 3 System Model

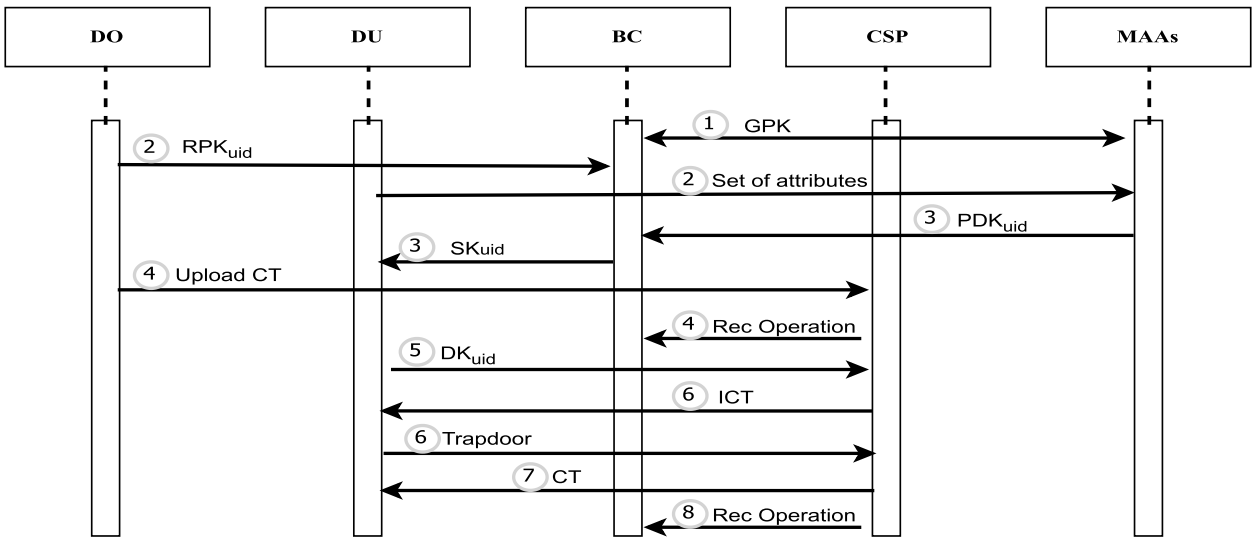

Fig. 4 Workflow

confirming the authorization for access and verifying the status of any revocation requests, the CS proceeds to compute the intermediate ciphertext. Subsequently, the CS transmits this intermediate ciphertext ICT in response to the DU. Following the recovery of secret parameters from the intermediate ciphertext (ICT), the DU generates a semantic-aware search trapdoor and submits it to the CS. Afterwards, given the trapdoor, the CS performs semantic searching and outputs the encrypted top-k related documents.

Smart contract

In the proposed scheme, two categories of smart contracts are introduced: *SystemContract* and *KeyGenContract*. During the Setup phase, *SystemContract* compiles partial attribute sets from various attribute authorities and employs Lagrange polynomial interpolation to construct global public parameters. This same approach enables attribute authorities to generate delegated keys by gathering partial delegated keys from each DU. *RecordContract* captures the user's identity

and the operations executed on CS over stored data for documentation.

Algorithm 1 SystemContract

Randomly selects elements α, β from $\mathbb{Z}_p$
if valid(sender,AA_{List}==true) **then**
  counter++;
  if counter $\geq$ threshold k **then**
    Compute $\prod_{i=1}^{k}(e(g,g)^{\alpha_i\beta_i})^{L_i}$ and $\prod_{i=1}^{k}((g^{\alpha_i})^{L_i}$
    Broadcast $g^{\alpha}, e(g,g)^{\alpha,\beta}$
  end
end
Computes $g_1 = g^{\alpha}$, $g_2 = g^{\beta}$, h^{β} and $g_3 = g^{\beta^2}$
Selects two polynomial $h(x)$ and $q(x)$ of degree n, subject to the constraint that $q_{(0)} = \beta$.
Define two publicly computable functions $T, V : \mathbb{Z}_p \rightarrow \mathbb{G}$; $T(x)$ maps to $g^{x^n}.g^{h(x)}$ and $V(x)$ maps to $g^{q(x)}$

Algorithm 2 KeyGenContract

if valid(sender,AA_{List}==true) **then**
  counter++;
  if counter $\geq$ threshold k **then**
    for each attribute $\in$ non-negative attribute **do**
      $K_i = (K_i^{(1)} = g_2^{t_{uid}\lambda_i}.T(x_i)^{r_{x_i}}, K_i'^{(1)} = g_2^{\lambda_i}.T(x_i)^{r_{x_i}}, K_i^{(2)} = g^{r_{x_i}})$
    end
    for each attribute $i \in$ negated attribute **do**
      $K_i' = (K_i^{(3)} = g_2^{t_{uid}\lambda_i + r_{x_i}}, K_i'^{(3)} = g_2^{\lambda_i + r_{x_i}}, K_i^{(4)} = V(x_i)^{r_{x_i}}, K_i^{(5)} = g^{r_{x_i}})$
    end
    Compute $DK = \prod_i^k (PDK)^{L_i}$
    Compute $RK = \{D_0 = g^{\alpha}.g^{\beta^2 t_{uid}}, D_1 = (g^{\beta.uid}h)^{t_{uid}}, D_2 = g^{-t_{uid}}\}$
  end
end
Send the SK=$\{ \mathcal{K}_{local} = (\frac{1}{t}), \mathcal{DK} = (\{K_i, K_i'\} \mid_{i \in S_{uid}}), RK\}$

Algorithm 3 RecordContract

Set up the RecordTable
// Introduce a fresh entry into the table
Entry entry = table.newEntry();
entry.set("Data Hash", V_i; "User ID", uid; "User Action", "Operation");

Proposed schemes construction

Concrete construction

1) **Setup**(λ): This algorithm is run by the attribute authorities and blockchain to provide a working environment for the proposed scheme. It initiates the system contract with the help of security parameters λ as input. Defines a bilinear group $\mathbb{G}$ of prime order p with generator g and h. It also select a universal set of attributes $U = \{x_1, x_2, \ldots, x_n\}$ and two random secret values $\{\alpha, \beta\} \in \mathbb{Z}_p^*$. It also generates a secret share for each attribute authority using Shamir's secret sharing scheme. Each attribute authority $AA_i \in AA_k$ computes the partial public parameters $\{e(g,g)^{\alpha_i\beta_i}, g^{\alpha_i}\}$ using their secret share and sends them to system contract. Upon receiving, the *SystemContract* uses the Lagrange polynomial interpolation to compute global public parameters.

Further, it randomly defines two polynomials $p(x)$ and $q(x)$ of degree n in a random manner, with the condition that $q(0) = \beta$. It then creates two functions, $U(x)$ and $V(x)$, which can be computed publicly and map to $g_2^{x^n} g^{p(x)}$ and $g^{q(x)}$, respectively. Notably, the use of the Lagrange Coefficient allows for the evaluation of $g^{p(x)}$ and $g^{q(x)}$ using the public key components [30]. Finally, set the MSK and the global public key components GPK.

Algorithfim 4 Setup

Input: λ.
Output: System public parameter PK and master secret key MSK.
Generate universal set of attributes U= $\{x_1, x_2, \ldots, x_n\}$.
Create a multiplicative cyclic group $\mathbb{G}$ with prime order p.
Using Shamir's secret sharing scheme generates partial public parameter $\{e(g,g)^{\alpha_i\beta_i}, g^{\alpha_i}\}$
Publish the global public parameters as GPK as $= (g, g_1, g_2, g_3, h^{\beta}, \{g^{q(i)} \mid i \in n\}, \{g^{h(i)} \mid i \in n\}, e(g,g)^{\alpha\beta})$
Accordingly, sets the $MSK = (\alpha, \beta)$

2) **Registration** : DO selects a random value $q_{uid} \in \mathbb{Z}_p$ and compute $RPK_{uid} = g^{q_{uid}}$. DO keeps the q_{uid} as confidential and transmits the RPK_{uid} to the blockchain. Subsequently, the consensus nodes activate the *Record-Contract* to record the DO who accessed the data.

3) **KeyGen**(**MSK**, **GPK**) : This phase runs by attribute authorities and KeyGenContract, binds access control components from the system attribute set U to the DU secret key components. This DU is then eligible for the decryption of the access control components if the attached attribute matches the secret key. For each attribute $x \in S_{uid}$, AA_i randomly select $r_{x_i} \in \mathbb{Z}_p$ and compute partial delegated key (PDK) components k_i, k_i' and send them to KeyGenContract. First, it obtains share λ_i for the system parameter α by applying the linear secret sharing mechanism Π and randomly selecting a value $t_{uid} \in \mathbb{Z}_p$. Further, using the Lagrange interpolation method, KeyGenContract combines the partially delegated key components as a SK and sends it to DU.

Algorithm 5 *KeyGen*

Input: GPK, and S.
Output: Partial delegation key $\mathcal{PDK}$ and local key $\mathcal{K}_{local}$
For each attribute $x_i \in S_{uid}$ the AA_i selects a random value $r_{x_i} \in \mathbb{Z}_p$
for each attribute $\in$ non-negative attribute **do**
  $K_i = (K_i^{(1)} = T(x_i)^{r_{x_i}}, K_i^{(2)} = g^{r_{x_i}})$
end
for each attribute $i \in$ negated attribute **do**
  $K_i' = (K_i^{(3)} = g_2^{r_{x_i}}, K_i^{(4)} = V(x_i)^{r_{x_i}}, K_i^{(5)} = g^{r_{x_i}})$
end
Sets $\mathcal{PDK} = (\{K_i, K_i'\} \mid_{i \in S_{uid}})$

4) **EncDoc**(**PK**, $\mathcal{K}$, **D**) : DO generates two keys. The first one, V, is an m-bit randomly generated configuration vector, while the second is $\mathcal{K} \in \mathbb{Z}_p$. The configuration vector V helps the DO scramble the Doc2Vec hidden layer weight matrix M_w and the secure inner product matrices. The key $\mathcal{K}$ is for document set D encryption. It also randomly generates two $M \times M$ dimensional invertible matrices and a random value $s \in \mathbb{Z}_p$. Then, the DO uses his own set of documents D to train the Doc2Vec model and gets the m-dimensional feature vector D_{v_i} for each document D_i in D. The reason for an own-trained neural network instead of a pre-trained neural network is to avoid the large dictionary that would produce a huge word vector for its dataset's vocabulary. Additionally, the dimension of the feature vector in the Doc2Vec model is much less than the words in the vocabulary of the document set. After the normalization, these feature vectors are treated as a semantic-aware plaintext index for its source documents. After which, the DO freezes the trained model and gets its hidden layer weights matrix M_w. Next, the DO encrypts the plaintext indexes D_{v_i} in D_v to its equivalent secure I_i in I using secure inner product operation as shown in Algorithm 6 in steps $6-10$. Also, encrypts each D_i in D, using symmetric key $\mathcal{K}$ to get its $\tilde{D}_i$ in $\tilde{D}$. The rows of the weights matrix M_w are the word vector representation for each word in the dictionary for our document set D. These normalized rows against each feature expose the semantic relationship for a given word. Hence, the DO uses the configuration vector V to obfuscate this semantic relationship, as depicted in Algorithm 6 in steps $14-18$. Similar operations are performed with the transformation matrix M_1 and M_2.

Let the revocation list $RL = \{RPK_{uid_1}, RPK_{uid_2}, \ldots RPK_{uid_n}\}$ of r users. The algorithm will split s into r random share $s_1, s_2, \ldots, s_r$ such that $\sum_{s_i} = s$ and compute C_0 and C_1 accordingly.

To dictate the access control through the set of negative and non-negative attributes $\gamma \in \mathbb{Z}_p^*$, DO computes the access control components Ac. Then, convert the configuration vector V into an integer I_v and encrypt it with symmetric key $\mathcal{K}$ for eligible DU. Further, the whole ciphertext components are set to CT. Finally, the DO uses SHA_{256} to compute the hash value $v_i = SHA(CT)$ and send $(CT, v_i, address_{DO})$ it to CS. Subsequently, the CS submits the $(v_i, CS_{id}, \text{"}upload\text{"})$ to the address of DO's *RecordContract*.

Algorithm 6 EncDoc

Input: System public parameters PK, Symmetric $\mathcal{K}$, Set of document D
Output: Ciphertext Components CT
DO randomly generates m-bit configuration vector V, $m \times m$ dimensional invertible matrices M_1 and M_2, secret s and symmetric key $\mathcal{K}$ for document set encryption
DO trained the Doc2Vec model with document set D, get m-dimensional feature vector D_{v_i} for each D_i in D
Freeze the own-trained Doc2Vec model and gets its hidden layer weight matrix M_w
for each $D_i \in$ D **do**
  if $V[j] = 1$ **then**
    $D'_{v_i}[j] = Rand()$ and $D''_{v_i}[j] = D_{v_i}[j] - D'_{v_i}[j]$
  end
  if V[j]=0 **then**
    $D'_{v_i} = D''_{v_i}[j] = D_{v_i}[j]$
  end
end
Compute secure index I_i
$I_i = \{D'_{v_i}.M_1^T, D''_{v_i}.M_2^T\}$ and $I = I \cup I_i$
Using symmetric key $\mathcal{K}$ encrypt D_i to its $\tilde{D}_i$ and $\tilde{D} = \tilde{D} \cup \tilde{D}_i$
for each $j \in V$ **do**
  if $V[j] = 1$ **then**
    $\tilde{M}_w = \tilde{M}_w \cup RoTR(M_w[Row_i])$
    $\tilde{M}_1 = \tilde{M}_1 \cup RoTR(M_1[Row_i])$
    $\tilde{M}_2 = \tilde{M}_2 \cup RoTR(M_2[Row_i])$
  end
end
Split s into r random share $s_1, s_2, \ldots, s_r$ such that $\sum_{s_i} = s$ for each revoked user in RL.
Compute $C_0 = g^{\beta s}$ and $C_1 = (C_{i,1} = g^{\beta s_i}, C_{i,2} = (g^{\beta^2.uid_i} h^{\beta}) \mid_{i \in r})$
Compute access control for $\gamma \subset \mathbb{Z}_p^*$ as $Ac = (\gamma, A^{(2)} = g^s, \{A_x^{(3)} = T(x)^s \mid_{x \in \gamma}\}\{A_x^{(4)} = V(x)^s \mid_{x \in \gamma}\})$
Convert configuration vector V into an integer I_v and encrypt it along with symmetric key $\mathcal{K}$ as $A = (\{A^{(0)} = \mathcal{K}.e(g_1, g_2)^s\}, \{A^{(1)} = I_v.e(g_1, g_2)^s\})$
The whole ciphertext components is given by $CT = (A, Ac, C_0, C_1, \tilde{M}_1, \tilde{M}_2, \tilde{M}_w, \tilde{D}, I)$

4) **Search$_A$**$(\mathcal{DK}, \mathbf{CT})$*:* Let the DU submits the $\mathcal{DK}$. This phase of the system model, as depicted in Algorithm 7, is run by CS and is divided into the following phases:

- *Access Verification:* CS first checks whether the attribute embedded into the $\mathcal{DK}$ satisfies the set of attributes attached to the access control components. If it does not satisfy, CS terminates the search phase; otherwise, it proceeds to process as follows: For each non-negated attributes x, CS computes $F_i = \frac{e\left(K_i'^{(1)}, A^{(2)}\right)}{e\left(K_i^{(2)}, A_i^{(3)}\right)}$. Similarly for negated attributes x', CS computes $F_i = \frac{e\left(K_i'^{(3)}, A^{(2)}\right)}{e\left(K_i^{(5)}, \prod_{x \in S}(A_x)^{(4)\sigma_x}\right).e\left(K_i^{(4)}, A^{(2)}\right)^{\sigma_{x_i}}}$ Finally, the verification is confirmed if $\prod_{i \in I} F_i^{w_i} = e(g,g)^{\alpha\beta}$, else $F_i = \perp$.
- *Revocation Identity:* Similarly, the CS checks whether the DU's RPK_{uid} is in the revocation RL list by simply computing

$$F_i = \frac{e(C_0, D_0)}{A.B} \tag{1}$$

Where $A = e\left(D_1, \prod_{i=1}^{r} C_{i,1}^{1/(RPK_{uid} - RPK_{uid_i})}\right)$ and $B = e\left(D_2, \prod_{i=1}^{r} C_{i,2}^{1/(RPK_{uid} - RPK_{uid_i})}\right)$ If $RPK_{uid} = RPK_{uid_i}$, the CS will fail to get two linearly independent equations and hence fails to solve the above equation for $e(g,g)^{s\alpha\beta}$.
- *Ciphertext Pre-computation:* If CT is accessible, this phase is similar to the above, except for two components $K_i'^{(1)}$ and $K_i'^{(3)}$ proceeds as follows: For each non-negated attribute x, the CS computes $IC_i = \frac{e\left(K_i^{(1)}, A^{(2)}\right)}{e\left(K_i^{(2)}, A_i^{(3)}\right)}$. Similarly, for negated attributes x', CS computes $IC_i = \frac{e\left(K_i^{(3)}, A^{(2)}\right)}{e\left(K_i^{(5)}, \prod_{x \in S}(A_x)^{(4)\sigma_x}\right).e\left(K_i^{(4)}, A^{(2)}\right)^{\sigma_{x_i}}}$ Further it computes $\prod_{i \in I} IC_i^{w_i} = e(g_2, g)^{ts\alpha}$ and set it to $\{I_R\}$. Finally, the CS returns the intermediate ciphertext $\tilde{CT}$ to the DU.

Algorithm 7 *Search$_A$*$(\mathcal{DK}, CT)$

Input: $\mathcal{DK}, CT$
Output: Precomputed ciphertext $\tilde{CT}$
for each x (non-negated attribute) in S **do**
  $F_i = \frac{e(K_i'^{(1)}, A^{(2)})}{e(K_i^{(2)}, A_i^{(3)})}$
end
for each x' (negated attribute) in S **do**
  $F_i = \frac{e(K_i'^{(3)}, A^{(2)})}{e(K_i^{(5)}, \prod_{x \in S}(A_x)^{(4)\sigma_x}).e(K_i^{(4)}, A^{(2)})^{\sigma_{x_i}}}$
end
if $\prod_{i \in I} F_i^{w_i} = e(g,g)^{s\alpha\beta}$ and $RPK_{uid} \neq RPK_{uid_i}$ **then**
  for each x (non-negative attribute) in S **do**
    $IC_i = \frac{e(K_i^{(1)}, A^{(2)})}{e(K_i^{(2)}, A_i^{(3)})}$
  end
  for each x' (negative attribute) in S **do**
    $IC_i = \frac{e(K_i^{(3)}, A^{(2)})}{e(K_i^{(5)}, \prod_{x \in S}(A_x)^{(4)\sigma_x}).e(K_i^{(4)}, A^{(2)})^{\sigma_{x_i}}}$
  end
  Compute $\prod_{i \in I} IC_i^{w_i} = e(g_2, g)^{ts\alpha}$ and set it to $\{I_R\}$
    $\tilde{CT} = (\{I_R\}, \tilde{M}_1, \tilde{M}_2, \tilde{M}_w, A^{(1)})$
else
  else $F_i = \perp$
end

5) **TDGen**($\mathcal{K}_{\mathbf{local}}$, **Q**,$\tilde{\mathbf{CT}}$)*:* Implemented in Algorithm 8 and run by the DU in our system model. Let Q be the query keyword set in the search trapdoor. DU first computes $e(g_2,g)^{\alpha s}$ to recover the integer I_v and subsequently convert it into configuration vector V. DU gets the un-permuted transformation matrices M_1, M_2 and weight matrix M_w by applying inverse shift row transformation using configuration vector V. Loads the Doc2vec model with M_w and inputting query keyword set Q, obtain the query feature vector Q_v with m-dimensional. Finally, using the secure inner product encryption operation to get trapdoor vector $\tilde{Q}_v$ and send it to the CS.

Algorithm 8 TrapGen

Input: $Q, k_{local}, \tilde{CT}$
Output: $\tilde{Q}_v$
DU compute $\{I_R\}^{\mathcal{K}_{local}} = e(g_2,g)^{\alpha s}$.
Compute I_v to recover the integer and convert it into configuration vector V .
$V = \frac{A^{(1)}}{e(g_2,g)^{\alpha s}}$
$= \frac{I_v \times e(g_1,g_2)^s}{e(g_2,g_1)^s}$
$= I_v$
for each $j \in V$ **do**
  if $V[j] = 1$ **then**
    $M_w = M_w \cup RoTL(\tilde{M}_w[row_i])$
    $M_1 = M_1 \cup RoTL(\tilde{M}_1[row_i])$
    $M_2 = M_2 \cup RoRL(\tilde{M}_2[row_i])$
  end
end
Loading the Doc2vec Model with trained weights matrix M_w and input query keywords Q, the DU gets the query vector Q_v with m-dimensional features.
for each $i \in V$ **do**
  if $V[i] = 0$ **then**
    $Q'_v[i] = Rand()$ and $Q''_v[i] = Q_v[i] - Q'_v[i]$.
  end
  if $V[i] = 0$ **then**
    $Q'_v[i] = Q''_v[i] = Q_v[i]$
  end
end
Compute secure search trapdoor $\tilde{Q}_v$ using M_1 and M_2 as
$\tilde{Q}_v = \{\tilde{Q}'_v, \tilde{Q}''_v\} = \{\tilde{Q}_v.M_1^{-1}, Q''_v.M_2^{-1}\}$

6) **Search**$_{\mathbf{B}}$($\tilde{\mathbf{Q}}_{\mathbf{v}}$, **I**)*:* For each document D_i, the CS performs the inner product operation between the secure index I_i in I and the trapdoor $\tilde{Q}_v$ to get the semantic-aware ranked scores as shown in the following equation. The CS invokes the *RecordContract* for the corresponding DO using their addresses. Then, it uploads the hash value of ciphertext V_i and data user's id to the smart contract as (V_i, *uid*, "*search*"). The CS, along with $A^{(0)}$, returns the top-k documents as a search result R_{score} to the DU.

$$\begin{aligned}\tilde{Q}_v.I_i &= \left\{Q'_v.M_1^{-1}, Q''_v.M_2^{-1}\right\}.\left\{D'_{v_i}.M_1^T, D''_{v_i}.M_2^T\right\} \\ &= Q'_v.M_1^{-1}.\left(D'_{v_i}.M_1^T\right)^T + Q''_v.M_2^{-1}.\left(D''_{v_i}.M_2^T\right)^T \\ &= Q'_v.D'_{v_i} + Q''_v.D''_{v_i} \\ &= Q_v.D_{v_i}\end{aligned} \tag{2}$$

7) **Decryption**($\mathcal{K}_{\mathbf{local}}$, $\mathbf{A^{(0)}}$)*:* DU computes the hash value of the ciphertext CT of ranked top-k received documents. If it is not the same, the algorithm halts. Otherwise send the (V_i, *uid*, "*decryption*") to *RecordCcontract* and recover the symmetric key $\mathcal{K}$ by computing

$$\begin{aligned}&= \frac{A^{(0)}}{e(g_2,g)^{\alpha s}} \\ &= \frac{\mathcal{K}.e(g_1,g_2)^s}{e(g_2,g)^{\alpha s}} \\ &= \frac{\mathcal{K}.e(g_1,g_2)^s}{e(g_1,g_2)^s} \\ &= \mathcal{K}\end{aligned}$$

Enhanced S^3*DBMS* scheme

We proposed an Enhanced S^3DBMS(ES^3DBMS) scheme to further strengthen our basic scheme's security. A detailed description of this is provided in the following subsections.

Security improvement

ES^3DBMS attains enhanced security by utilizing the learning with errors (LWE)-based secure kNN algorithm to encrypt features indices [31]. This approach guarantees strong privacy protection for the underlying feature vectors. The required changes in the ES^3DBMS are as follows:

EncDoc(**PK**, $\mathcal{K}$, $\mathcal{M}$): The DO generates a set of encryption keys for feature vectors, represented as γ, M, M^{-1}. Here, γ is a publically randomly chosen integer from the set $\mathbb{Z}_{p_1}$, while M is a randomly generated invertible matrix with dimensions $2m \times 2m$. M^{-1} represents the inverse of matrix M. Additionally, the integers p_1 and p_2 define the range of numbers, with p_1 significantly greater than p_2. Next, to encrypt the features vectors, the DO extends the m-dimensional vectors to $2m$-dimensional as

$$V_v = \left\{v_{i,1}, v_{i,2}, \ldots, v_{i,d}, \frac{1}{2}\sum_{l=1}^{m} v_{i,j}, \alpha\right\}$$

where $\alpha \in \mathbb{Z}_{p_2}^{d-1}$ are selected by DO as random numbers for each feature vector $v_{i,j}$. Further, DO encrypt each extended feature vector $v_{i,j}$ as

$$E_m(v_i) = (\gamma.v_i + \epsilon_i)M = \tilde{I}_i$$

Here, ϵ_i represent a random integer noise vector, $\gamma >> 2|max(\epsilon_i)|$ represents the absolute value of elements in ϵ_i.

TDGen$(\tilde{\mathbf{CT}}, \mathbf{Q}, \mathcal{K}_{\mathbf{local}})$*:* Similarly, the DU extends the query vector Q_v to $Q_v = (\delta_j q_1, \delta_j q_2, \ldots, \delta_j q_m, \delta_j, \beta_j)$

where, $\delta_i \in \mathbb{Z}_{p_2}$, $\beta_j \in \mathbb{Z}_{p_2}^{m-1}$ are radom numbers. Further, DU encrypts this extended query vector Q_v as

$$E_{M^{-1}}(Q_v) = M^{-1}\left(\gamma.Q_v^T + \epsilon_j^T\right) = \tilde{Q}_v$$

where, $\epsilon_j \in Z_{p_1}^{2m}$ random integer noise vector.

Search$_{\mathbf{B}}(\tilde{\mathbf{I}}, \tilde{\mathbf{Q}}_{\mathbf{v}})$*:* With the extended dimension, the relevance score between the trapdoor $\tilde{Q}_v$ and every document index in $\tilde{I}$ is computed as

$$\begin{aligned}
&= E_M(\tilde{I_v}).E_{M^{-1}}(\tilde{Q}_v) \\
&= \tilde{I}_v M.M^{-1}\tilde{Q}_v \\
&= -\frac{\delta_j}{2}\sum_{k=1}^{n}\left(I_v^2 - 2I_vQ_v\right) + \alpha\beta_j^T
\end{aligned}$$

Correctness of access control

Algorithm *Search*$_A$ tells us that the CS can confirm the access authorization by checking whether the below equation is true or not:

$$\Pi_{i\in I}F_i^{w_i} = e(g,g)^{s\alpha\beta} \tag{3}$$

The set of authorized entities is denoted as $\tilde{\gamma} = N(\gamma) \in \mathbb{A}$. Let I be the set of indices i such that $\widetilde{x} \in \tilde{\gamma}$ and $w_i \in Z_p | i \in I$ be a collection of constants. A valid share λ_i of secret share α based on the $\prod$ protocol satisfies the equation $\Sigma_{i\in I} w_i\lambda_i = \alpha$. The CS calculates this equation for each non-negated attribute $x_i \in \gamma$ (i.e., $\widetilde{x}_i \in \gamma'$).

$$\begin{aligned}
F_i &= \frac{e\left(K_i'^{(1)}, C^{(2)}\right)}{e\left(K_i^{(2)}, C_i^{(3)}\right)} \\
&= \frac{e\left(g_2^{\lambda_i}.T(x_i^{r_i}), g^s\right)}{e(g^{r_i}, T(x_i)^s)} \\
&= \frac{e(g_2,g)^{s\lambda_i} \times e(T(x_i)^{r_i}, g^s)}{e(g^{r_i}, T(x_i)^s)} \\
&= \frac{e(g_2,g)^{s\lambda_i} \times e(T(x_i), g)^{r_i s}}{e(T(x_i), g)^{r_i s}} \\
&= e(g_2,g)^{s\lambda_i}
\end{aligned}$$

Similarly for negated attributes $x_i \notin \gamma$ (so $\widetilde{x}_i \in \gamma'$), we let $\gamma_i = \gamma \cup \{x_i\}$ and compute Lagrange interpolation over the points in set γ_i to get the coefficient $\{\sigma_x\}_{x\in\gamma_i}$, such that $\sum_{x\in\gamma_i}\sigma_x q(x) = q(0) = \beta$. Now CS computation proceeds as follows:

$$\begin{aligned}
F_i &= \frac{e\left(K_i'^{(3)}, C^{(2)}\right)}{e\left(K_i^{(5)}, \prod_{x\in S}(C_x)^{(4)\sigma_x}\right).e\left(K_i^{(4)}, C^{(2)}\right)^{\sigma_{x_i}}} \\
&= \frac{e\left(g_2^{\lambda_i + r_i)}, g^s\right)}{e\left(g^{r_i}, \prod_{x\in\gamma}(V(x)^s)^{\sigma_x}\right).e\left(V(x_i)^{r_i}, g^s\right)^{\sigma_{x_i}}} \\
&= \frac{e\left(g_2^{\lambda_i}.g_2^{r_i}, g^s\right)}{e(g^{r_i}, \prod_{x\in\gamma}(V(x)^s)^{\sigma_x}).e(V(x_i)^{r_i}, g^s)^{\sigma_{x_i}}} \\
&= \frac{e(g_2,g)^{s\lambda_i} \times e(g_2,g)^{r_i s}}{e\left(g^{r_i}, g^{s\Sigma_{x\in\gamma}\sigma_x q(x)}\right).e\left(g^{r_i\sigma_{x_i}q(x_i)}, g^s\right)} \\
&= \frac{e(g_2,g)^{s\lambda_i} \times e(g,g)^{r_i s\beta}}{e(g,g)^{r_i s\Sigma_{x\in\gamma'}\sigma_x q(x)}} \\
&= \frac{e(g_2,g)^{s\lambda_i} \times e(g,g)^{r_i s\beta}}{e(g,g)^{r_i s\beta}} \\
&= e(g_2,g)^{s\lambda_i}
\end{aligned}$$

Finally,

$$\begin{aligned}
\Pi_{i\in I}F_i^{w_i} &= e(g_2,g)^{s\alpha} \\
\Pi_{i\in I}F_i^{w_i} &= e(g,g)^{s\alpha\beta}
\end{aligned}$$

Security analysis

For the access control threat, the security proof is modeled in the form of a security game between an attacker $\mathcal{A}$ and challenger $\mathcal{C}$.

Theorem 1 *If a probabilistic time adversary (PPT) can break the access control components in Algorithm* 2 *with advantage* ϵ *in the selective-set models. We can construct a simulator* $\mathcal{B}$ *to the decisional BDH game with the advantage* $\frac{\epsilon}{2}$.

Proof

$\mathcal{B}$ plays the role of challenger $\mathcal{C}$ which randomly select $a, b, c, z \in \mathbb{Z}_p^*$ flips a binary coin $\mu \in \{0, 1\}$, outside of $\mathcal{B}$ view. If $\mu = 0$, $\mathcal{C}$ sets $\mathbb{Z}_\mu = e(g,g)^{a,b,c}$, otherwise it sets $\mathbb{Z}_\mu = e(g,g)^z$.

Init. The simulator $\mathcal{B}$ runs $\mathcal{A}$. $\mathcal{A}$ declares a revocation list $RL = \{RPK_{uid_1}, RPK_{uid_2}, \ldots RPK_{uid_n}\}$ along with corrupted attribute authorithies $AA_c \subseteq AA$ and chooses the challenge access structure γ of d attributes.

Setup. $\mathcal{B}$ now creates the public keh components by assigning $g^{\alpha} = A$ and $g^{\beta} = B$ by implicitly assigning $\alpha = a$ and $\beta = b$. However, for revocation key RK it will set β as $b_1 + b_2 + \ldots b_r$. It then randomly chooses $y \in \mathbb{Z}_p$, a polynomial $f(x)$ of degree d randomly and fixes a degree d polynomial $U(x)$ as per the following procedure:

For all $x \in \gamma$, sets $u(x) = -x^d$, otherwise sets $u(x) \neq -x^d$. $\mathcal{B}$ implicitly set the polynomial h and q as follows: First, sets $h(x) = \beta u(x) + f(x)$. It then, randomly selects points $\theta_{x_1}, \theta_{x_2}, \ldots, \theta_{x_d} \in \mathbb{Z}_p$ for a set $\gamma = \{x_1, x_2, \ldots, x_d\}$ and sets $q_{(x_i)} = \theta_{x_i}$ such that $q(0) = \beta$. Finally, it sends the public key components $\{\{g^{q_i} \mid_{i\in[1,d]}\}, g^{h_i} = g_2^{u_i} g^{f_i} \mid_{i\in[1,d]}, \ g^{\beta} = \prod_{i\in RL} g^{b_i}, \ g^{\beta^2} = \prod_{i,j\in RL} g^{b_i.b_j}, h = \prod_{i\in RL}(g^{b_i - t_{uid}} g^{y})\}$ to $\mathcal{A}$.

Phase 1. $\mathcal{A}$ repeatedly asks for a number of access control structures except γ. Suppose $\mathcal{A}$ asks for a secret key components such that $\tilde{\mathbb{A}}(\gamma) = 0$, where $\tilde{\mathbb{A}}$ is early defined as $NM(\mathbb{A})$ for some monotonic access structure $\mathbb{A}$ over a set of attributes S, for some linear secret sharing scheme Π. To map the secret shares to negated or non-negated attributes, we let M be the sharing matrix over Π, then the sharing for this simulation is as follows:

First, we set the secret $\alpha = a$ for this distribution then, we randomly select a vector $v = (v_1, v_2, \ldots, v_{n+1}) \in \mathbb{Z}_p^{n+1}$ since $(1, 0, \ldots, 0)$ is independent of $M_{\gamma'}$. Therefore, we can efficiently compute [32] a vector $\mathcal{W} = (\mathcal{W}_1, \ldots, \mathcal{W}_{n+1})$ such that $(1, 0, \ldots, 0).\mathcal{W} = \mathcal{W}_{\infty} = 1$ and $M_{\gamma'}.\mathcal{W} = \overrightarrow{0}$, where $M_{\gamma'}$ is the sub-matrix of M associated with a set of attributes in γ'. We now define a uniformly distributed vector $\upsilon = v + (a - v_1)\mathcal{W}$ subject to the constraint that $\upsilon_1 = a$. To compute the shares $\lambda = M_{\upsilon}$, we have $\lambda_i = M_i \upsilon = M_i v$ for all $x_i \in \gamma'$, such that it was no dependence on a. First, for negated attributes $\breve{x}_i = x_i'$, we show how to compute the secret key components. Note that if $x_i \in \gamma$ if and only if $x_i \notin \gamma$.

- The secret share λ_i may depend linearly on a if and only if $x_i \in \gamma$. $\mathcal{B}$ selects r_i' and $t \in \mathbb{Z}_p$ at random and set $\mathcal{K}_{local} = (\frac{1}{t})$. Now, by letting $q(x_i) = \theta_{x_i}$ and $r_i = -t\lambda_i + r_i'$, $\mathcal{B}$ outputs the following valid delegated key components. $K_i' = \left(K_i^{(3)} = g_2^{r_i}, K_i^{(4)} = g^{\theta_{x_i}(-t\lambda_i + r_i')}, K_i^{(5)} = g^{-t_{\lambda_i} + r_i'}\right)$.
- The secret share λ_i is independent on a if and only if $x_i \in \gamma$ and hence known to the $\mathcal{B}$. In this case, the simulator picks $r_i \in \mathbb{Z}_p$ and randomly outputs the following delegated key components: $K_i = \left(K_i^{(3)} = g_2^{t\lambda_i + r_i}, K_i^{(4)} = V(x_i)^{r_i}, K_i^{(5)} = g^{r_i}\right)$

Now, for non-negated attribute $\breve{x}_i = x_i$, we describe how to compute the secret key components.

- The secret share λ_i is independent of any secret, if and only if $x_i \in \gamma$. In this case, $\mathcal{B}$ picks $r_i \in \mathbb{Z}_p$ and randomly output $K_i = \left(K_i^{(1)} = g^{t\lambda_i}.T(x_i)^{r_i}, K_i^{(2)} = g^{r_i}\right)$
- The secret share λ_i depends on a if and only if $x_i \notin \gamma$. Then $\mathcal{B}$ let $g_3 = g^{\lambda_i}$ and select $r_i' \in \mathbb{Z}_p$ at random and output the component $K_i^{(1)}$ and $K_i^{(2)}$ of delegated key K_i as

$$K_I^{(1)} = g_3^{\frac{-f(x)}{x_i^d + u(x_i)}} \left(g_2^{x_i^d + u(x_i)}.g^{f(x)}\right)^{r_i'}$$

$$K_i^{(2)} = g_3^{\frac{-1}{x_i^d + u(x_i)}} g^{r_i'}$$

Now $\mathcal{B}$ constitute secret key for identies in RL. For each $RPK_{uid} \in RL$, $\mathcal{B}$ select a random $Z_i \in Z_p$ and set the t_{uid} implicitly as $t_{uid} = -ai^2 + 2i$. The secret key components for RPK_{uid} is computed as:

$$D_0 = \prod_{1\leq j,k\leq n, j\neq i} \left(g^{-ab_j b_k / b_i^2}\right) \prod_{1\leq j,k\leq n} \left(g^{ab_k z_i}\right) D_1 = \left(\prod_{1\leq j\leq n, j\neq i} \left(g^{-a.b_j/b_i^{2(RPK_i - RPK_j)}}\right) g^{\left((RPK_i - RPK_j).b_j z_i\right)}\right) g^{(-a/b_i^2)^y} g^{yz_i} D_2 = g^{a/b_i^2} g^{-z_i}$$

Challenge. Adversary $\mathcal{A}$ select two equal lenght messages M_1 and M_2 and submit it to $\mathcal{B}$ for encryption. $\mathcal{B}$ randomly chooses $s', s_1', \ldots s_r' \in Z_p$ such that $s' = \sum_i s_i'$. Now $\mathcal{B}$ flips a fair coin $v \in \{0, 1\}$ and returns the encryption of M_v as:

$$AC = \left(\gamma, C_0 = g^s g^{s'}, C_{i,1} = g^{sa_i} \left(\prod_j g^{a_j}\right)^{s_i'}, C_{i,2} = \left(\prod_{1\leq j\leq r^*} \left(g^{sa_i a_j}\right)^{RPK_i - RPK_j}\right) \left(g^{a_i s}\right)^y u_i^{s_i'}, C^{(1)} = M \times Z,\right.$$
$$\left. C^{(2)} = C, \left\{C_x^{(3)} = C^{f(x)} \mid_{x\in\gamma}\right\}, \left\{C_x^{(4)}) = C^{x} \mid_{x\in\gamma}\right\}\right)$$

Now the following cases arise:

- If $\mu = 0$, then $Z = e(g,g)^{abc}$. Then by inspection, the encryption of Ac is valid encryption under the set γ.
- if $\mu = 1$, then $Z = e(g,g)^{z}$. Since z is random, the ciphertext $C^{(1)} = M_{\upsilon} \times e(g,g)^{z}$ will be a random element of $\mathbb{G}_T$ from $\mathcal{A}$ point of view and hence contain no information about M_{ν}.

Phase 2. $\mathcal{B}$ acts the same way as it did in Phase 1.
Guess. $\mathcal{B}$ will submit a guess ν' of ν. There are two probabilities, either $\mu' = 1$, which indicates it has given a valid BDH-tuple, or $\mu' = 0$, which indicates a random 4-tuple. Now, the probability analysis is given as:

- In the case where $\mu = 1$, $\mathcal{A}$ gains no information about ν. Therefore, we have

$$Pr[\mu = 1 \mid \nu' \neq \nu] = \frac{1}{2} \tag{4}$$

Since the simulator guesses $\mu' = 1$ when $\nu \neq \nu'$ also, we have

$$Pr[\mu' = \mu \mid \mu = 1] = \frac{1}{2} \tag{5}$$

- if $\mu = 0$, then $\mathcal{A}$ sees the encryption of M_{ν}. The $\mathcal{A}$ advantage is ϵ by assumption. Therefore, we have

$$Pr[\mu = 0 \mid \nu' = \nu] = \frac{1}{2} + \epsilon \tag{6}$$

Since the simulator guess $\mu' = 0$ when $\nu = \nu'$ so we gain

$$Pr[\mu' = \mu \mid \mu = 0] = \frac{1}{2} + \epsilon \tag{7}$$

Using Eqs. (4) and (6), the overall advantage of the algorithm $\mathcal{B}$ in the decisional BDH game is

$$\begin{aligned} &\mid \frac{1}{2} Pr[\mu' = \mu | \mu = 0] + \frac{1}{2} + Pr[\mu' = \mu | \mu = 1] - \frac{1}{2} \mid \\ &= \mid [\frac{1}{2}(\frac{1}{2} + \varepsilon) + \frac{1}{2}.\frac{1}{2}] - \frac{1}{2} \mid = \frac{\varepsilon}{2} \end{aligned} \tag{8}$$

Performance analysis

This section thoroughly evaluates our proposed approach in different scenarios, focusing on functionality, precision, and time cost. The experiments were carried out on a computer with the Windows 7 operating system, featuring 32 GB RAM with two 64-bit 3.4 GHz Intel(R) Core(TM) i7-3770 CPUs using Python in the Gensim framework, and trained it on a dataset of 20 newsgroups with 11,315 articles, but without GPUs. It is expected that the use of GPUs would significantly speed up both the training and testing processes. In our experiments, 'm,' 'h,' 'n,' and 'k' represent the number of features, keywords, total documents, and required documents, respectively. Default values are in Table 1.

Functionality analysis

This section compares our proposed scheme's functionality with secure semantic searching schemes and blockchain-based searchable encryption schemes. As demonstrated in Table 2, our scheme supports semantic term matching constraints (STMC) in M/M settings and is more comprehensive than existing schemes. Our scheme and the schemes proposed by [20, 24, 33, 34] utilize blockchain technology to ensure reliable search outcomes. The previous schemes lack support for semantic-aware search, which means they only allow exact-keyword matching over encrypted indices. As a result, they are not suitable for practical application as they fail to learn the latent semantic intention of the user search query for a better search experience. Our scheme produces a probabilistic query for document retrieval. A unique query feature vector is generated every time for the same set of keywords to hide the search pattern. Compared with schemes, our scheme uses a deep learning doc2vec model that has led to the representation of more powerful predictive word embedding that not only captures the semantic features of words but also surrounding context [29, 35]. Unlike the mentioned schemes, the proposed scheme puts access control on the outsourced document, which also provides users with revocations. Hence, it is more suitable for a broader range of applications. Similar to [33] scheme, our scheme employs semantic term-matching constraints to enhance search accuracy. The remaining blockchain-based schemes rely on single or multi-keyword matching to identify documents that contain the exact keywords in the indices. Consequently, these schemes focus solely on determining if the retrieved documents possess a specific keyword, disregarding any consideration of semantic matching between index and query keywords. As a result, these schemes fail to meet the fundamental retrieval heuristic STMC-1. According to STMC-2, the retrieval algorithm should prioritize a document that exactly matches the query words with a higher relevance

Table 1 Default parameters

Parameters	m	n	h	k
Values	300	6000	5	10

Table 2 Comparison of functionalities

	Blockchain-based schemes				Semantic-aware schemes				Our
	[24]	[33]	[34]	[20]	[36]	[37]	[8]	[9]	
Semantic retrieval	×	✓	×	×	✓	✓	✓	✓	✓
Probabilistic	×	×	×	×	✓	✓	×	✓	✓
Retrieval heuristic	×	✓	×	×	×	×	×	×	✓
Weight model	–	YAKE	–	–	Word2Vec	SBERT	–	Word2Vec	Doc2Vec
Multi-keyword	×	✓	×	✓	✓	✓	✓	✓	✓
Result ordering	×	✓	×	✓	✓	✓	✓	✓	✓
Integrity	✓	✓	✓	✓	×	×	✓	×	✓
No TTP	✓	✓	✓	✓	×	×	×	×	✓
Access control	×	×	×	×	×	×	×	×	✓
Policy expressibility	–	–	–	–	×	×	×	×	✓
Revocation	×	×	×	×	×	×	×	×	✓

score than documents that only contain words related in meaning. While semantic similarity is valuable for document retrieval, relying too much on it is not recommended. However, the previous semantic-based searching schemes fail to acknowledge this constraint. Our proposed scheme introduces the truncated cosine similarity metric $\widehat{cos} = (\tilde{I}_{v_i}, \tilde{Q}_v)$ to balance the semantic matching and exact matching effectively. STMC-3 requires a damping effect in accumulating relevant score processes of specific query words, enabling more distinct query words to contribute to the search process. However, the mentioned semantic searching schemes do not fulfill this requirement. The scheme in [9, 36] word2vec operates at a word-level granularity, disrespecting the context or order in which words appear in a query. This limitation prevents the fine-grained control necessary for introducing a damping effect and prioritizing specific query words. Compared to these schemes, the doc2vec distributed document embedding reduce the impact of specific terms by assigning different dimensions to different aspect of the document, ensuring a balanced representation. Additionally, the weight of the trained DO model is securely transferred to the DU side to facilitate relevant query feature vector formulation.

Our scheme, along with previous approaches [8, 9, 36, 37], enables semantic search on encrypted documents. Compared to these schemes, our proposed scheme stands alone in the domain of semantic searching by integrating essential features of multi-attribute authority, revocation, and access control simultaneously. Although, the scheme in [8] supports only revocation. In terms of the weight model, it is noteworthy that schemes [9, 36] employ Word2Vec. Word2Vec is primarily designed to capture semantic relationships among individual words by representing them as dense vector embedding. However, relying solely on Word2Vec may not adequately capture the holistic meaning of an entire document. In contrast, our scheme utilizes Doc2Vec, representing the entire document as a fixed-length vector. This characteristic of Doc2Vec makes it a more appropriate choice for document-level semantic similarity tasks. By considering the document as a whole, Doc2Vec enables a more comprehensive understanding of the document's semantic meaning.

Semantic precision evaluation

We leverage the categorizations inherent in the 20 newsgroup dataset, employing them as 22 discrete semantic classes that encompass a range of subjects, including but not limited to baseball, politics, and hardware. Furthermore, we utilize the articles within these classes to assess the semantic precision of our proposed scheme quantitatively. To achieve a level of semantic precision that is reasonably valid, we make the underlying assumption that the documents belonging to different classes hold no significant semantic relevance to each other. The data user's query is expected to contain keywords from the same class. For instance, if the query includes terms like "pitch," "helmet," and "National League," the documents retrieved from the class associated with baseball would be considered accurate results. We use the semantic precision calculation method used in [38] to evaluate the search precision of our scheme as follows:

$$P = \frac{TP}{TP + FP} \times 100\% \tag{9}$$

Where TP and FP denote the quantities of true positive and false positive returned documents, respectively.

In this evaluation of semantic precision, we omitted expansion-based schemes. Despite their ability to

retrieve the top-k documents for a given search query, these schemes rely on shallow semantic relationships, falling short of capturing the underlying latent semantic nuances of the search query. Schemes [9, 36, 37], similar to our approach, build indices and query vectors utilizing word/document embedding to retain the semantic relationships among words. This is why these schemes have been chosen for comparing their precision. Among the word-level embedding schemes, Scheme [9] exhibits the lowest performance. This can be attributed to its approach of accumulating word-level embedding in a bitwise manner, leading to the creation of compact word vectors for representing documents. Consequently, this method falls short of capturing complete semantic information within the word vectors. Much like Scheme [9], Scheme [37] employs word-level embeddings for constructing feature vectors. However, in this scheme, an additional step is taken where the k-means clustering algorithm is utilized to categorize each document prior to generating the feature vectors. This approach results in the construction of indices based on category-document vectors, contributing to its enhanced precision compared to Scheme [9]. Both our proposed approach and the method outlined in [36] show noticeable improvements in precision. However, our scheme outperforms the approach in [36]. This advantage can be attributed to several aspects of our scheme. We address the problem of excessive matching by incorporating a truncated cosine similarity measure to assess the semantic similarity between embedding. Additionally, our approach utilizes doc2vec, a technique more suitable for creating embedding at the document level as compared to word2vec. Doc2vec captures both the context in which words appear within a document and the distinct identity of the document itself, contributing to its enhanced performance. We further illustrate the effectiveness of precision by varying the parameters n, k, and h for these schemes. As depicted in Fig. 5(a) and (b), it becomes evident that the our scheme exhibits a higher level of semantic search precision in comparison to the remaining word2vec-based schemes. The average search precision for our scheme is observed to be approximately 88.3% and 87.2% when the parameter k is assigned values of 50 and 100, respectively. In Fig. 5(c) and (c), when n is set at 4000 and 8000, our approach achieves average search precision rates of 86.7% and 86.4%, respectively. Furthermore, as k increases, both our scheme and word2vec-based schemes experience a decline in precision. This stems from the limited number of relevant documents within each semantic class in the dataset. The elevated k introduces irrelevant search documents into the result list, deviating from the user's query intention. The different h settings also maintain the same order of precision as observed with n. Both Fig. 5(e) and (f) show that our scheme achieves an average semantic precision of about 86.9% and 87.6% for different n settings. The wrod2vec and doc2vec models, which accurately capture the underlying meaning of documents, are only slightly affected by the increase in the query keyword h in terms of its precision.

Experimental analysis of S^3BDMS

Scheme [39], alongside our proposed framework, employs attribute-based access control, blockchain technology, and revocation mechanisms, thereby eliminating the need for a singular trusted entity. Consequently, a comprehensive simulation is executed to evaluate the operational efficacy of these schemes in relation to encryption and decryption algorithms. Both schemes are executed using the Java Pairing-Based Cryptography library, employing an 80-bit elliptic curve group generated from the equation $y^2 = x^3 + x$, operating over a 256-bit finite field. Figure 6(a) displays the time it takes for encryption function in both schemes. This helps us understand how changing attributes affect the schemes. In our system, we see a sudden increase in time at the start of creating the secure index, as shown in Fig. 6(a). This increase results from the splitting process and matrix multiplications involved in secure kNN inner product operations. Moreover, the only factor that appears to influence how long the encryption process lasts is the number of attributes present in the access control structure. Regarding decryption, as illustrated in Fig. 6(b), the time taken by the DU is less than the scheme in [39] because most of the intense computation operations, i.e., bilinear pairing, from the DU end are outsourced to the computationally rich cloud server with the help of the delegated key component (DK) generated by the attribute authorities. As a result, in decryption, a constant number of operations are allocated for the DU to decrypt the ciphertext.

Conclusion and future work

This paper introduces a semantic search scheme based on deep learning and multi-attribute authority within a multi-user setting in cloud storage infrastructure. We present an innovative perspective regarding attribute-based encryption and transfer learning within the context of the SE framework. We use attribute-based encryption to securely transfer trained parameters from the DO to the DU, enabling the creation of a query feature vector within the model's training feature space to obtain highly accurate ranked results. Concurrently, blockchain's smart contracts enable a multi-attribute authority to generate user private keys and system-wide parameters through consensus in the context of mutual distrust within an M/M setting. Moreover, the scheme's flexibility is improved by incorporating non-monotonic

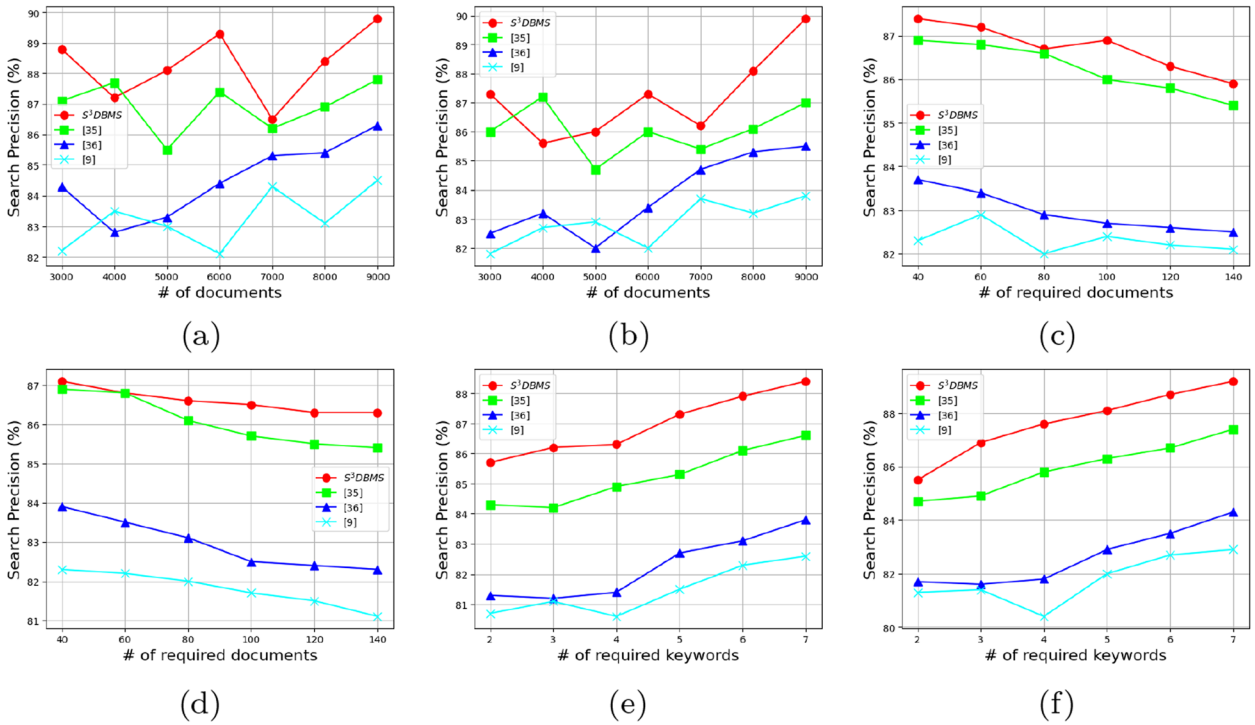

Fig. 5 Semantic precision versus n, (**a**) k=50, (**b**) k=100. Semantic precision versus k, (**c**) n=4000, (**d**) n=800. Semantic precision versus h, (**e**) n=4000, (**f**) n=8000. Semantic precision versus m, (g)

access structures and direct revocation. Users' activities are transparently and reliably recorded on the blockchain through smart contracts.

Although we're using a pre-trained neural network, fine-tuning this model with the data owner proves to be resource-intensive, especially for devices with limited resources at the user's end. In our future work, we aim to explore privacy-preserving outsourced machine learning techniques. Our focus will be on outsourcing the extraction of feature vectors to a powerful cloud server for the neural network model. This exploration aims to tackle the challenges posed by resource constraints at the user end and improve the privacy aspects of machine learning processes.

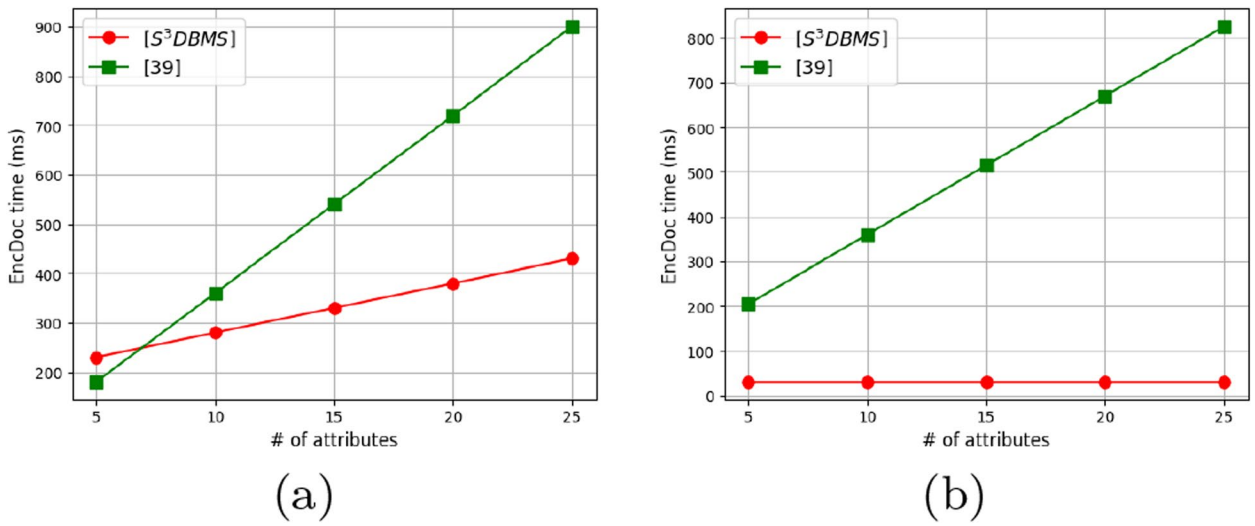

Fig. 6 Time cost in each algorithm. **a** EncDoc, (**b**) TrapGen and Decryption

Authors' contributions
Shahzad Khan formulated the research concept, participated in its design, and made substantial contributions to the manuscript's writing. Haider Abbas shared valuable knowledge about the theory, carefully improved the research methods, and oversaw the paper while providing important feedback at every stage. Muhammad Binsawad oversaw the implementation, conducted the simulations, and interpreted the results.

Funding
This research work was funded by Institutional Fund Projects under grant no. (IFPIP: 208-611-1443). The authors gratefully acknowledge technical and financial support provided by the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Availability of data and materials
In this paper, we utilized real-world dataset of 20 newsgroups, publicly available with 20 different categories.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 29 September 2023 Accepted: 19 December 2023
Published online: 30 January 2024

References

1. Sun X, Zhu Y, Xia Z, Chen L (2014) Privacy-preserving keyword-based semantic search over encrypted cloud data. Int J Secur Appl 8(3):9–20
2. Xia Z, Zhu Y, Sun X, Chen L (2014) Secure semantic expansion based search over encrypted cloud data supporting similarity ranking. J Cloud Comput 3:1–11
3. Fu Z, Sun X, Linge N, Zhou L (2014) Achieving effective cloud search services: multi-keyword ranked search over encrypted cloud data supporting synonym query. IEEE Trans Consum Electron 60(1):164–172
4. Moh TS, Ho KH (2014) Efficient semantic search over encrypted data in cloud computing. In: 2014 International Conference on High Performance Computing & Simulation (HPCS), IEEE, pp 382–390
5. Fu Z, Wu X, Wang Q, Ren K (2017) Enabling central keyword-based semantic extension search over encrypted outsourced data. IEEE Trans Inf Forensic Secur 12(12):2986–2997
6. Fu Z, Xia L, Sun X, Liu AX, Xie G (2018) Semantic-aware searching over encrypted data for cloud computing. IEEE Trans Inf Forensic Secur 13(9):2359–2371
7. Wong WK, Cheung DWl, Kao B, Mamoulis N (2009) Secure knn computation on encrypted databases. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, pp 139–152
8. Yang W, Zhu Y (2020) A verifiable semantic searching scheme by optimal matching over encrypted data in public cloud. IEEE Trans Inf Forensic Secur 16:100–115
9. Liu Y, Fu Z (2019) Secure search service based on word2vec in the public cloud. Int J Comput Sci Eng 18(3):305–313
10. Chai Q, Gong G (2012) Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers. In: 2012 IEEE international conference on communications (ICC), IEEE, pp 917–922
11. Stefanov E, Papamanthou C, Shi E (2013) Practical dynamic searchable encryption with small leakage. Cryptol ePrint Arch
12. Zhu J, Li Q, Wang C, Yuan X, Wang Q, Ren K (2018) Enabling generic, verifiable, and secure data search in cloud services. IEEE Trans Parallel Distrib Syst 29(8):1721–1735
13. Li J, Wu J, Jiang G, Srikanthan T (2020) Blockchain-based public auditing for big data in cloud storage. Inf Process Manag 57(6):102382
14. Jing N, Liu Q, Sugumaran V (2021) A blockchain-based code copyright management system. Inf Process Manag 58(3):102518
15. Zhao Q, Chen S, Liu Z, Baker T, Zhang Y (2020) Blockchain-based privacy-preserving remote data integrity checking scheme for iot information systems. Inf Process Manag 57(6):102355
16. Campanile L, Iacono M, Marulli F, Mastroianni M (2021) Designing a gdpr compliant blockchain-based iov distributed information tracking system. Inf Process Manag 58(3):102511
17. Chen Q, Srivastava G, Parizi RM, Aloqaily M, Al Ridhawi I (2020) An incentive-aware blockchain-based solution for internet of fake media things. Inf Process Manag 57(6):102370
18. Hong H, Sun Z (2021) A secure peer to peer multiparty transaction scheme based on blockchain. Peer Peer Netw Appl 14:1106–1117
19. Hong H, Hu B, Sun Z (2021) An efficient and secure attribute-based online/offline signature scheme for mobile crowdsensing. Hum-Cent Comput Inf Sci 11:26
20. Hu S, Cai C, Wang Q, Wang C, Wang Z, Ye D (2019) Augmenting encrypted search: a decentralized service realization with enforced execution. IEEE Trans Dependable Secure Comput 18(6):2569–2581
21. Jiang S, Cao J, McCann JA, Yang Y, Liu Y, Wang X, Deng Y (2019) Privacy-preserving and efficient multi-keyword search over encrypted data on blockchain. In: 2019 IEEE international conference on Blockchain (Blockchain), IEEE, pp 405–410
22. Li H, Gu C, Chen Y, Li W (2019) An efficient, secure and reliable search scheme for dynamic updates with blockchain. In: Proceedings of the 2019 the 9th International Conference on Communication and Network Security, pp 51–57
23. Tahir S, Rajarajan M (2018) Privacy-preserving searchable encryption framework for permissioned blockchain networks. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber. Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, pp 1628–1633
24. Cai C, Weng J, Yuan X, Wang C (2018) Enabling reliable keyword search in encrypted decentralized storage with fairness. IEEE Trans Dependable Secure Comput 18(1):131–144
25. Ostrovsky R, Sahai A, Waters B (2007) Attribute-based encryption with non-monotonic access structures. In: Proceedings of the 14th ACM conference on Computer and communications security, pp 195–203
26. Fang H, Tao T, Zhai C (2004) A formal study of information retrieval heuristics. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp 49–56
27. Fang H, Zhai C (2005) An exploration of axiomatic approaches to information retrieval. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp 480–487
28. Guo J, Fan Y, Ai Q, Croft WB (2016) Semantic matching by non-linear word transportation for information retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp 701–710
29. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781
30. Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24, Springer, pp 457–473
31. Chen C, Zhu X, Shen P, Hu J, Guo S, Tari Z, Zomaya AY (2015) An efficient privacy-preserving ranked keyword search method. IEEE Trans Parallel Distrib Syst 27(4):951–963
32. Anton H, Rorres C (2013) Elementary linear algebra: applications version. John Wiley & Sons
33. Yang W, Sun B, Zhu Y, Wu D (2021) A secure heuristic semantic searching scheme with blockchain-based verification. Inf Process Manag 58(4):102548
34. Li J, Li D, Zhang X (2023) A secure blockchain-assisted access control scheme for smart healthcare system in fog computing. IEEE Internet Things J
35. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543

36. Chen L, Xue Y, Mu Y, Zeng L, Rezaeibagha F, Deng RH (2022) Case-sse: Context-aware semantically extensible searchable symmetric encryption for encrypted cloud data. IEEE Trans Serv Comput 16(2):1011–1022
37. Hu Z, Dai H, Liu Y, Yang G, Zhou Q, Chen Y (2022) Csmrs: An efficient and effective semantic-aware ranked search scheme over encrypted cloud data. In: 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), IEEE, pp 699–704
38. Gabryel M, Damaševičius R, Przybyszewski K (2018) Application of the bag-of-words algorithm in classification the quality of sales leads. In: Artificial Intelligence and Soft Computing: 17th International Conference, ICAISC 2018, Zakopane, Poland, June 3-7, 2018, Proceedings, Part I 17, Springer, pp 615–622
39. Wang X, Zhou Z, Luo X, Xu Y, Bai Y, Luo F (2021) A blockchain-based fine-grained access data control scheme with attribute change function. 2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing. Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI), IEEE, pp 348–356

Publisher's Note